

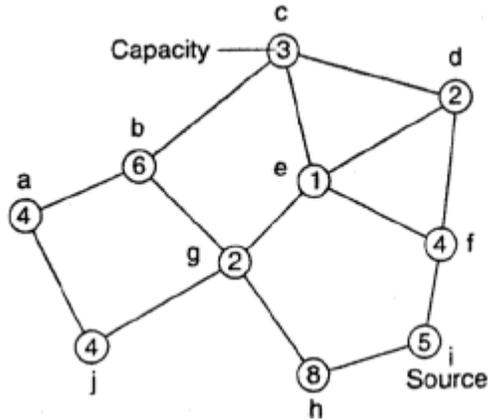
UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Sistemas Distribuídos 2019.1

Profa.: Simone Martins

Gabarito Lista 3

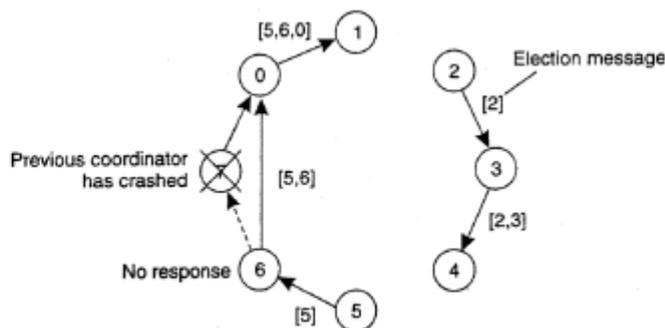
1. Considere o comportamento de duas máquinas em um sistema distribuído. Ambas possuem relógios que devem gerar um pulso 1000 vezes por milissegundo. Um deles efetivamente gera mas o outro gera um pulso somente 990 vezes por milissegundo. Se os dois recebem atualização da hora UTC uma vez por minuto, calcule o desajuste entre os relógios.
Resposta: O segundo relógio gera 990,000 pulsos por segundo, acarretando um erro de 10 mseg por segundo. Em um minuto esse erro cresce para 600 ms. Outra maneira de ver isso é que o segundo relógio é um por cento mais lento, então depois de um minuto ele fica defasado de $0.01 \times 60 \text{ s}$, ou 600 ms.
2. Explique a diferença entre multicast totalmente ordenado e ordenado por causalidade.
Resposta: No multicast totalmente ordenado todos os processos recebem as mensagens na mesma ordem. No multicast ordenado por causalidade, cada mensagem que é causada por uma outra deve ser recebida depois da mensagem que causou o envio em todos os processos. Mas as mensagens que não tem relação de causalidade podem ser recebidas em diferentes ordens pelos processos.
3. Explique o princípio do algoritmo de Berkeley para ajustar relógios em sistemas distribuídos.
Resposta: No algoritmo Berkeley o servidor consulta todas as máquinas de tempos em tempos, obtendo o horário de cada máquina, gera uma média de todas as horas, e informa a todos os computadores o deslocamento de tempo a ser feito.
4. Marcas de tempo de Lamport não capturam causalidade. Explique.
Resposta: Na verdade marcas de tempo de Lamport capturam causalidade, mas não capturam totalmente. Caso um evento b ocorra por causa de um evento a, a marca de tempo de Lamport será menor para a do que para b. Mas se a marca de relógio de a for menor que a de b, isso não implica que a causou b.
5. Compare os algoritmos de exclusão mútua Centralizado, Descentralizado, Distribuído, e Token Ring em termos de desempenho e problemas existentes.
Resposta: Centralizado: Vantagens: Simples, fácil de entender e de implementar; É justo (segue o FCFS) Não há inanição (starvation); Desvantagens: Como todo sistema centralizado possui um ponto de erro e um ponto de gargalo; Processos não conseguem distinguir coordenador inativo de permissão negada, caso se o recurso não está disponível nenhuma mensagem é enviada.
Distribuído: Vantagens: Torna solução centralizada original menos vulnerável a falhas; Desvantagens: Risco de conceder permissão dupla ao recurso que gerencia.
Distribuído: Vantagens: Exclusão mútua é garantida; Não há deadlock nem starvation; Desvantagens: Ponto de falha único foi substituído por n pontos de falha; Trocou 1 gargalo por n gargalos
Token Ring: Vantagens: Garante a exclusão mútua; Não há deadlock nem starvation. Desvantagens: A dificuldade é recuperar uma ficha perdida.
6. Suponha que dois processos detectem a morte do coordenador simultaneamente e ambos decidam convocar uma eleição que utilize o algoritmo do valentão. O que acontecerá?
Resposta: Cada um dos processos de numeração mais alta receberá duas mensagens ELECTION, mas ignorarão a segunda. A eleição prosseguirá como de costume.
7. Considere a rede sem fio abaixo. Descreva como ocorre a eleição de um líder utilizando o algoritmo de eleição em rede sem fio visto em sala de aula supondo que o nó que inicia o processo é o nó i.



Resposta:

- i envia eleição para f e h
- f e h tem i como pai e h envia eleição para g e f envia eleição para d e e
- g tem h como pai e d e e tem f como pai
- g envia eleição para j, b, e e e envia eleição para c, g e d e d envia eleição para e e c
- e responde com ACK para g e d e d responde com ACK para e
- b e j tem g como pai
- c tem e como pai e envia ACK para d
- b envia eleição para c e a
- j envia eleição para a
- c envia eleição para b
- a tem j como pai, envia ACK para b e envia valor (a,4) para j
- j envia valor (j,4) para g
- c envia ACK para b e envia valor (c,3) para e
- b envia valor (b,6) para g
- g envia valor (b,6) para h
- c envia valor (c,3) para e, que envia valor (c,3) para f
- d envia valor (d,2) para f
- f envia valor (f,4) para i
- h envia valor (h,8) para i
- i resolve o que h será o coordenador (h,8)

8. Há duas mensagens de eleição circulando simultaneamente na figura abaixo. Isso acarreta em problemas? Proponha uma solução para eliminar uma das mensagens sem afetar o algoritmo.



Resposta:

Não faz mal ter as duas mensagens, mas seria mais elegante se alguma pudesse ser retirada. Um algoritmo para retirar a mensagem é descrito a seguir: Quando um processo recebe uma mensagem ELEIÇÃO, ele verifica quem a iniciou. Se ele próprio a iniciou (ou seja, seu número está no topo da lista), ele transforma a mensagem em uma mensagem COORDENADORA. Se não iniciou nenhuma mensagem ELEIÇÃO, adiciona o seu número de processo e o encaminha ao longo do anel. No entanto, se ele enviou sua própria mensagem ELEIÇÃO anteriormente e acaba de descobrir uma concorrente, ele compara o número do processo do processo origem da mensagem com o seu próprio. Se o outro processo tiver um número menor, ele descartará a mensagem em vez de transmiti-la. Se o competidor for maior, a mensagem é encaminhada da maneira usual. Dessa forma, se várias mensagens ELECTION forem iniciadas, aquela com a primeira entrada mais alta sobrevive. O resto é retirado ao longo da rota.